

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application.

1. (currently amended) A method for providing content, comprising the steps of:
receiving a request from a user device for particular content, said request is received at a server;
accessing a mark-up language description of said particular content, said mark-up language description includes one or more source files which describe a behavior of said particular content on a user interface of said user device, said particular content includes data for rendering on said user interface, said one or more source files define a connection to an external data source for said data, said external data source is external to said server;
accessing said data at said external data source based on said ~~mark-up language description~~ one or more source files which define said connection to said external data source, said server performs said accessing;
compiling said mark-up language description of said particular content, including said data, to create executable code for a ~~rendering entity~~ said user device, said step of compiling is performed at said server in response to said request; and
transmitting said executable code from said server to said user device, said user device provides said particular content via said user interface according to said one or more source files when said user device rendering entity executes said executable code.
2. (cancelled)
3. (cancelled)
4. (currently amended) A method according to claim 1, wherein:

said user device includes a rendering entity and a browser, said rendering entity is a plug-in to said a-browser, said plug-in is embedded in said browser before said request, and said rendering entity executes said executable code.

5. (previously presented) A method according to claim 1, wherein:

after said data is accessed from said external data source, said data is provided in a markup language document, said step of compiling includes converting said data in said markup language document to ActionScript and compiling said ActionScript into ActionScript byte code.

6. (original) A method according to claim 1, wherein:

said step of transmitting includes using HTTP to transmit said executable code via a network.

7. (currently amended) A method according to claim 1, ~~further comprising the step of~~ wherein:

executing said executable code at said rendering entity, said rendering entity is a plug-in to a browser, said request for particular content is received from a is made by said browser in which a rendering entity is present as a plug-in to said browser, said browser is at said user device, and said rendering entity executes said executable code.

8. (currently amended) A method according to claim 1, further comprising the steps of:

accessing media content, said particular content includes said media content;

~~transforming said media content to an accepted format;~~

providing a reference ~~to said transformed media content~~ in said mark-up language description to a media file which contains said media content, said media file is external to said mark-up language description; executable code; and

~~adding said transformed media content to said executable code, said transformed media content is not compiled~~

transmitting said media file with said executable code from said server to said user device, a rendering entity at said user device renders said media content on said user interface when said media file is referenced when said executable code is executed.

9. (previously presented) A method according to claim 1, wherein said step of compiling comprises the steps of:

converting said mark-up language description to ActionScript; and
compiling said ActionScript into ActionScript byte code.

10. (currently amended) A method according to claim ~~8~~ 9, further comprising: ~~the steps of:~~

~~accessing media content, said particular content includes said media content;
transforming said media file content to an accepted format before said transmitting of said media file, said transforming is separate from said compiling; and adding said transformed media content to said executable code, said request is from a client associated with said rendering entity, said executable code implements said user interface, said user interface provides access to said particular content.~~

11. (previously presented) A method according to claim 1, further comprising the step of:

authenticating said request, said steps of compiling and transmitting are only performed if said step of authenticating is successful, different types of authenticating are provided for at least one of: a) different types of content and b) each item of content.

12. (cancelled)

13. (currently amended) A method according to claim 1, further comprising the steps of:
receiving a request at said server from said user device ~~a client associated with said rendering entity~~ for second content, said particular content includes a first application, said second content includes a second application called by said first application;

accessing a mark-up language description of said second content;
compiling said mark-up language description of said second content; and

transmitting said compiled mark-up language description of said second content from said server to said client user device.

14. (previously presented) A method for providing content, comprising the steps of:
receiving a request for particular content, said request is received at a server;
in response to said request, accessing first code associated with said particular content, said first code includes a mark-up language description and a scripting language description;
compiling said mark-up language description and said scripting language description to create combined executable code from both said mark-up language description and said scripting language description that implements a user interface that provides access to said particular content, said step of compiling is performed at said server in response to said request; and
transmitting said executable code from said server to a client.

15. (original) A method according to claim 14, wherein:
said request is from said client.

16. (original) A method according to claim 14, wherein:
said particular content includes data; and
said data is compiled to executable code during said step of compiling.

17. (original) A method according to claim 16, wherein:
said step of compiling includes converting said data to action script and compiling said action script into action script byte code.

18. (cancelled)

19. (previously presented) A method according to claim 14, wherein:
said markup language description includes elements which are identified by markup language tags, at least one of said elements provides a script source of said scripting language description.

20. (currently amended) A method according to claim 14, further comprising the steps of:

accessing media content, said particular content includes said media content;

transforming said media content to an accepted format, said transforming is separate from said compiling;

providing a reference to said transformed media content in said executable code; and

~~adding said transformed media content to said executable code, said transformed media content is not compiled~~

transmitting said transformed media content with said executable code from said server to said client for execution by said client, where, during the execution at said client, when said reference is reached, said client renders said transformed media content.

21. (previously presented) A method for providing content, comprising the steps of:

receiving a request for content that includes data other than code, said data is for rendering on a user interface at a client, and said request is received at a server;

accessing a mark-up language description associated with said content at said server, said mark-up language description defines a connection to an external data source for said data, said external data source is external to said server;

acquiring said data from said external data source in response to said mark-up language description, said data is acquired by said server;

compiling said content at said server to create executable code, said content is based on said mark-up language description and said data, said executable code includes a representation of said data, said step of compiling is performed in response to said request; and

transmitting said executable code from said server to said client.

22. (original) A method according to claim 21, wherein:

said request is from said client.

23. (previously presented) A method according to claim 21, wherein:

said executable code implements said user interface, said user interface provides access to said data.

24. (original) A method according to claim 21, wherein:

said step of compiling includes converting said data to action script and compiling said action script into action script byte code.

25. (cancelled)

26. (cancelled)

27. (currently amended) A method according to claim 21, further comprising the steps of:

accessing media content;

transforming said media content to an accepted format, said transforming is separate from said compiling; and

~~adding said transformed media content to said executable code, said transformed media content is not compiled~~

transmitting said transformed media content with said executable code from said server to said client.

28. (currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising the steps of:

receiving a request for particular content from a browser, said browser having a plug-in embedded therein, said request is received at a server;

accessing a mark-up language description of said particular content, said mark-up language description references a media file comprising at least one of audio, video and a movie;

compiling said mark-up language description of said particular content to create executable code for said a plug-in to said a browser, said executable code provides said particular content, said step of compiling is performed at said server in response to said request; and

transmitting said executable code and said media file from said server to said plug-in, said plug-in renders said particular content based on said executable code and said media file, ~~is not compiled.~~

29. (cancelled)

30. (original) One or more processor readable storage devices according to claim 28, wherein:

said executable code implements a user interface that provides access to said particular content.

31. (original) One or more processor readable storage devices according to claim 28, wherein:

said particular content includes data; and

said data is compiled to executable code during said step of compiling.

32. (currently amended) One or more processor readable storage devices according to claim 28, wherein said method further comprises the steps of:

transforming said media file accessing media content, said particular content includes said media content, transforming said media content to an accepted format before transmitting said media file, said transforming is separate from said compiling, and adding said transformed media content to said executable code.

33. (currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising the steps of:

receiving a request for particular content, said request is received at a server from a web client in which a plug-in is embedded;

accessing first code associated with said particular content;

in response to said request, compiling said first code to create executable code for said a plug-in to said web client, said executable code implements a user interface that provides access to said particular content, said step of compiling is performed at said server in response to said request; and

transmitting said executable code from said server to said plug-in for execution by said plug-in.

34. (cancelled)

35. (cancelled)

36. (currently amended) One or more processor readable storage devices according to claim 33, wherein said method further comprises the steps of:

accessing media content, said particular content includes said media content;

transforming said media content to an accepted format, said transforming is separate from said compiling; and

adding said transformed media content to said executable code.

37. (currently amended) One or more processor readable storage devices having processor readable code embodied on said processor readable storage devices, said processor readable code for programming one or more processors to perform a method comprising:

receiving a request for content that includes data other than code, said data is for rendering on a user interface by a rendering entity which is present at a client, said rendering entity is separate from a browser but operates within said browser, and said request is received at a server;

acquiring said data from a data source external to said server, said acquiring is performed by said server;

compiling said data at said server to create executable code for said rendering entity, said executable code includes a representation of said data, said step of compiling is performed in response to said request; and

transmitting said executable code from said server to said rendering entity.

38. (original) One or more processor readable storage devices according to claim 37, wherein:

said request is from said client.

39. (previously presented) One or more processor readable storage devices according to claim 37, wherein:

said executable code implements said user interface, said user interface provides access to said data.

40. (currently amended) One or more processor readable storage devices according to claim 37, wherein said method further comprises the steps of:

accessing media content;

~~transforming said media content to an accepted format; and~~

~~adding said transformed media content to said executable code~~

providing a reference to said media content in said executable code; and

transmitting said media content with said executable code from said server to said rendering entity.

41. (currently amended) An apparatus, comprising:

one or more storage devices; and

one or more processors in communication with said one or more storage devices, said one or more processors receive a request for particular content from an HTTP client having a plug-in embedded therein, said request is received at a server, ~~said request is from a client~~, said one or more processors access a mark-up language description of said particular content and compile said mark-up language description of said particular content to create executable code for said a plug-in to said

a HTTP client, said executable code provides said particular content, said compiling is performed at said server in response to said request, and said ~~one or more processors transmit said executable code~~ is transmitted from said server to said plug-in for execution by said plug-in.

42. (original) An apparatus according to claim 41, wherein:
said executable code implements a user interface that provides access to said particular content.

43. (original) An apparatus according to claim 41, wherein:
said particular content includes data; and
said data is compiled to executable code during said step of compiling.

44. (currently amended) An apparatus according to claim 41, wherein:
said particular content includes ~~media content~~ at least one of audio, video and a movie.

45. (currently amended) An apparatus, comprising:
one or more storage devices; and
one or more processors in communication with said one or more storage devices, said one or more processors perform a method comprising the steps of:

receiving a request for particular content, said request is received at a server, said request is from a client which, ~~said client~~ includes a browser and a rendering engine that is different than said browser but operates in connection with said browser;

accessing first code associated with said particular content at said server, said first code comprises elements that are identified by markup language tags, at least one of said elements references a source external to said server;

compiling said first code to create executable code for said rendering engine, said executable code implements a user interface that provides access to said particular content, said step of compiling is performed at said server in response to said request, and

transmitting said executable code from said server to said client for rendering of said particular content by said rendering engine.

46. (previously presented) An apparatus according to claim 45, wherein:
said particular content includes data stored at said source, said accessing first code includes accessing said data at said source; and
said data is compiled to executable code during said step of compiling.

47. (currently amended) An apparatus according to claim 45, wherein said method further comprises the steps of:

accessing media content, said particular content includes said media content, at least one of said elements identifies said media content;

transforming said media content to an accepted format, said transforming is separate from said compiling; and

transmitting adding said transformed media content with to said executable code to said client for rendering of said transformed media content by said rendering engine.

48. (previously presented) An apparatus, comprising:
one or more storage devices; and
one or more processors in communication with said one or more storage devices, said one or more processors: a) receive a request for content that includes data other than code, said request is received at a server, said request is from a client, b) access a mark-up language description and a scripting language description associated with said content at said server and acquire said data from a source external to said server, said data is acquired by said server, script code of said scripting language description is contained within script tags of said markup language description c) compile said mark-up language description and said scripting language description at said server to create executable code, said executable code includes a representation of said data, said compiling is performed in response to said request, and d) transmit said executable code from said server to said client.

49. (original) An apparatus according to claim 48, wherein:
said executable code implements a user interface that provides access to said data.

50. (previously presented) An apparatus according to claim 48, wherein:
said data includes media content.
51. (previously presented) A method according to claim 21, wherein:
said data is media data.
52. (currently amended) A method according to claim 4, wherein:
said request is received at said server from said user device and includes an indication that identifies a type of said rendering entity from a group of rendering entities; and
said compiling includes creating said executable code specific for said type of rendering entity in response to said indication.
53. (previously presented) A method according to claim 1, wherein:
said executable code comprises one or more binary files.
54. (previously presented) A method according to claim 1, wherein:
said executable code comprises at least one of object code and byte code.
55. (previously presented) One or more processor readable storage devices according to claim 33, wherein:
said first code comprises elements which are identified by markup language tags.
56. (previously presented) One or more processor readable storage devices according to claim 55, wherein:
at least one of said elements defines a view template of a user interface element, said view template is instantiated when said executable code is executed by said rendering entity.
57. (currently amended) One or more processor readable storage devices according to claim 56, wherein:

said elements comprise at least one element which defines a view class which supplies default properties, behavior, and child views which the view template instantiates, the child views are associated with a parent view.

58. (previously presented) One or more processor readable storage devices according to claim 55, wherein:

at least one of said elements references a media file comprising at least one of audio, video and a movie.

59. (cancelled)

60. (previously presented) One or more processor readable storage devices according to claim 55, wherein:

at least one of said elements references a media file that contains an animation.

61. (previously presented) One or more processor readable storage devices according to claim 55, wherein:

at least one of said elements references a media file that contains a movie.

62. (previously presented) One or more processor readable storage devices according to claim 28, wherein:

said media file comprises a .SWF file, said markup language description references said .SWF file.

63. (cancelled)

64. (previously presented) One or more processor readable storage devices according to claim 55, wherein:

at least one of said elements provides an inline definition of formatted text.

65. (currently amended) One or more processor readable storage devices according to claim 55, wherein:

at least one of said elements provides an inline definition of vector graphics;

66. (cancelled)

67. (previously presented) A method according to claim 1, wherein:
said markup language description comprises elements which are identified by markup language tags; and
said elements comprise at least one element which references said connection to said external data source.

68. (currently amended) One or more processor readable storage devices according to claim 55, wherein:

at least one of said elements defines a connection to a web service which is external to said server.

69. (previously presented) A method according to claim 1, wherein:
said compiling comprises parsing said markup language description to obtain first and second types of elements, providing said first and second types of elements to first and second compiling modules, respectively, to obtain first and second object code, respectively, and assembling said first and second object code into a single executable.

70. (previously presented) A method according to claim 69, wherein:
said first type of element defines at least one of a visual appearance of said particular content and a behavior of said particular content, and said second type of element defines said connection to said external data source.

71. (cancelled)

72. (cancelled)

73. (previously presented) A method according to claim 4, wherein:
said rendering entity is a Flash player.

74. (previously presented) A method according to claim 14, wherein:
at least one of said elements of said markup language description instantiates a class defined
in the scripting language description.

75. (previously presented) A method according to claim 14, wherein:
said scripting language description extends a class defined in said markup language
description.

76. (previously presented) A method according to claim 27, wherein:
said accepted format comprises at least one of a JPEG format and a GIF format.

77. (previously presented) One or more processor readable storage devices according to
claim 55, wherein:
said elements comprises elements which define script code, said script code specifies a visual
appearance of said user interface.

78. (previously presented) One or more processor readable storage devices according to
claim 55, wherein:
said elements comprises elements which define script code, said script code specifies an
application logic of said mark-up language description.

79. (previously presented) One or more processor readable storage devices according to
claim 55, wherein:

said elements comprises elements which define script code, said script code specifies a connection to an external data source, said external data source includes data for rendering on said user interface by said plug-in.

80. (previously presented) A method according to claim 28, wherein:
said plug-in is a Flash player.

81. (new) A method according to claim 8, further comprising:
providing an object in the executable code which identifies a name and/or format of the media content, the name and/or format is provided via the user interface when said media content is rendered.

82. (new) A method according to claim 8, wherein:
said request for particular content is received from a browser in which a plug-in to said browser is present, said browser is at said user device, and said plug-in renders said media content.